Bioconductor's Computational Ecosystem for Genomic Data Science in Cancer

Marcel Ramos[0000−0002−3242−0582], Lori Shepherd[0000−0002−5910−4010], Nathan Sheffield[0000−0001−5643−4068], Alexandru Mahmoud[0000−0002−3779−492X], Hervé Pagès[NA], Jen Wokaty[NA], Dario Righelli[0000−0002−2687−9928], Davide Risso[0000−0001−8508−5012], Sean Davis[0000−0002−8991−6458], Sehyun Oh[0000−0002−9490−3061], Levi Waldron[0000−0003−2725−0694], Martin Morgan[0000−0002−5874−8148], and Vincent Carey[0000−0003−4046−0063]

Marcel Ramos
City University of New York School of Public Health, New York, NY

Lori Shepherd
Roswell Park Comprehensive Cancer Center, Buffalo, NY

Nathan Sheffield
University of Virginia, Charlottesville, VA

Alexandru Mahmoud
Channing Division of Network Medicine, Mass General Brigham, Harvard Medical School, Boston, MA

Hervé Pagès
Fred Hutchinson Cancer Center, Seattle, WA

Jen Wokaty
City University of New York School of Public Health, New York, NY

Dario Righelli
Department of Statistical Sciences, University of Padova, Padova, Italy

Davide Risso
Department of Statistical Sciences, University of Padova, Padova, Italy

Sean Davis
University of Colorado Anschutz School of Medicine, Aurora, CO

Sehyun Oh
City University of New York School of Public Health, New York, NY

Levi Waldron
City University of New York School of Public Health, New York, NY

Martin Morgan
Roswell Park Comprehensive Cancer Center, Buffalo, NY

Vincent Carey
Channing Division of Network Medicine, Mass General Brigham, Harvard Medical School, Boston, MA

Abstract The Bioconductor project enters its third decade with over two thousand packages for genomic data science, over 100,000 annotation and experiment resources, and a global system for convenient distribution to researchers. Over 60,000 PubMed Central citations and terabytes of content shipped per month attest to the impact of the project on cancer genomic data science. This report provides an overview of cancer genomics resources in Bioconductor. After an overview of Bioconductor project principles, we address exploration of institutionally curated cancer genomics data such as TCGA. We then review genomic annotation and ontology resources relevant to cancer and then briefly survey analytical workflows addressing specific topics in cancer genomics. Concluding sections cover how new software and data resources are brought into the ecosystem and how the project is tackling needs for training of the research workforce. Bioconductor's strategies for supporting methods developers and researchers in cancer genomics are evolving along with experimental and computational technologies. All the tools described in this report are backed by regularly maintained learning resources that can be used locally or in cloud computing environments.

1 Introduction

Computation is a central component of cancer genomics research. Tumor sequencing is the basis of computational investigation of mutational, epigenetic and immunologic processes associated with cancer initiation and progression. Numerous computational workflows have been produced to profile tumor cell transcriptomes and proteomes. New technologies promise to unite sequence-based characterizations with digital histopathology, ultimately driving efforts in molecule design and evaluation to produce patient-centered treatments.

Bioconductor is an open source software project with a 20 year history of uniting biostatisticians, bioinformaticians, and genome researchers in the creation of an ecosystem of data, annotation, and analysis resources for research in genome-scale biology. This paper will review current approaches of the project to advancing cancer genomics. After a brief discussion of basic principles of the Bioconductor project, we will present a "top down" survey of resources useful for cancer bioinformatics. Primary sections address

- how to explore institutionally curated cancer genomics data
- genomic annotation resources relevant to cancer genomics
- analytical workflows
- components for introducing new data or analyses
- pedagogics and workforce development.

Two concluding offer discussion of possible paths forward, and a detailed description of software components underlying an exemplary integrative analysis of response to immune checkpoint blockade.

2 Bioconductor principles

2.1 R packages and vignettes

Software tools and data resources in Bioconductor are organized into "R packages". These are collections of folders with data, code (principally R functions), and documentation following a protocol specified in the Writing R Extensions manual [1]. R packages have a DESCRIPTION file with metadata about package contents and provenance. Package structure can be checked for validity using the `R CMD check` facility. Documentation of code and data can be programmatically checked for existence and validity. The DESCRIPTION file for a package specifies its version and also gives precise definition of how an R package may depend upon versions of other packages.

At its inception, Bioconductor introduced a new approach to holistic package documentation called "vignette". Vignettes provide narrative and explanation interleaved with executable code describing package operations. While R function manual pages describe the operation of individual functions, vignettes illustrate the interoperation of package components and provide motivation for both package design but also context for its use.

2.2 R package repositories; repository evolution

Bioconductor software forms a coherent ecosystem that can be checked for consistency of versions of all packages available in a given installation of R. Bioconductor packages may specify dependency on other Bioconductor packages, or packages that are available in the CRAN repository. Bioconductor does not include packages with dependencies on "github-only" packages. Later in this paper we will provide details on package quality assurance that provide a rationale for this restriction.

Major updates to the R language occur annually, and updates are preceded by careful assessment of effects of language change on Bioconductor package operations. These effects can be identified through changes in the output of R CMD check. The Bioconductor ecosystem is updated twice a year, once to coincide with update to R, and once about six months later. The semianual updates reflect the need to track developments in the fast-moving field of genomic data science.

2.3 Package quality assessment; installation consistency

The BiocCheck function is used to provide more stringent assessment of package compliance with basic principles of the Bioconductor ecosystem.

The BiocManager package provides for installing and updating package and has functionality for verifying the coherence and version status of the currently installed package collection. This is important in the context of a language and package ecosystem that changes every six months, while analyses may take years to complete. Tools for recreating past package collections are available to assist in reproducing outputs of prior analyses.

2.4 Unifying assay and sample data: SummarizedExperiment and MultiAssayExperiment

Most of the data from genome-scale experiments to be discussed in this chapter are organized in special data containers rooted in the concepts of the SummarizedExperiment class. Briefly, assay data are thought of as occupying a $G \times N$ array, and sample level data occupy an $N \times K$ table. The array and the table are linked together in the SummarizedExperiment; see Figure 1.

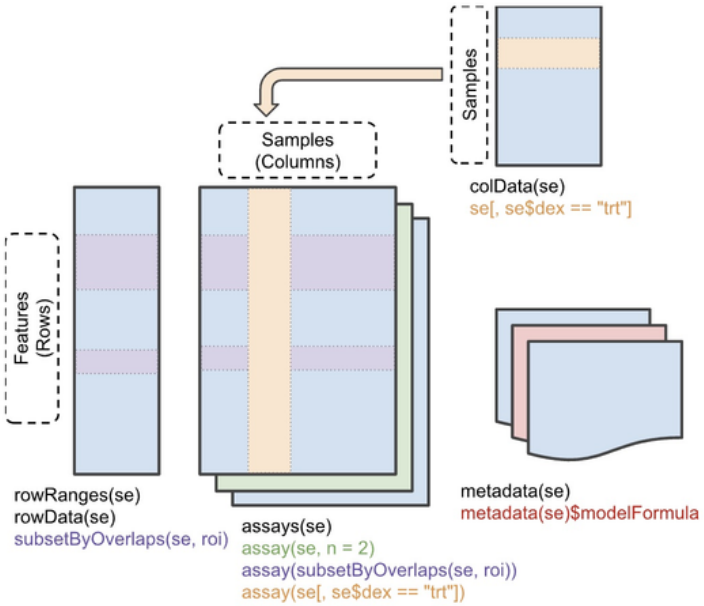


Fig. 1 SummarizedExperiment schematic.

Multiple representations of assay results may be managed in this structure, but all assay arrays must have dimensions $G \times N$.

For experiment collections in which the same samples are subjected to multiple genome-scale assays, MultiAssayExperiment containers are used. See Figure 2 for the layout.

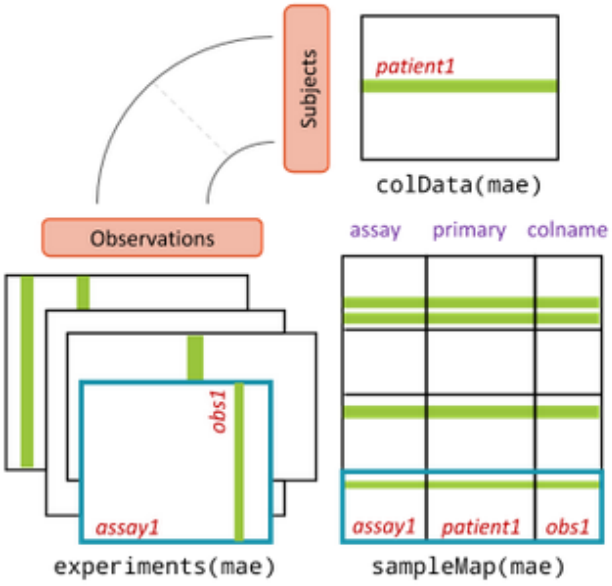


Fig. 2 MultiAssayExperiment schematic.

Further details on these data structures will be provided in section 6.

2.5 Downloading and caching cancer genomics data and annotations

Downloading and managing data from various online resources can be excessively time consuming. Bioconductor encourages data caching for increased efficiency and reproducibility. The caching data methods employed in Bioconductor allow analysis code to concisely refer to data resources as needed, with minimal attention to how data are stored, retrieved or transformed. It allows for easy management and reuse of data that are on remote servers or in cloud, storing source location and providing information for data updates. The BiocFileCache Bioconductor package handles data management from within R.

BiocFileCache is a general-use caching system but Bioconductor also provides "Hubs", AnnotationHub and ExperimentHub, to help distributed annotation or ex-

perimental data hosted externally. Both AnnotationHub and ExperimentHub use BiocFileCache to handle download and caching of data.

AnnotationHub provides a centralized repository of diverse genomic annotations, facilitating easy access and integration into analyses. Researchers can seamlessly retrieve information such as genomic features, functional annotations, and variant data, streamlining the annotation process for their analyses.

ExperimentHub extends this concept to experimental data. It serves as a centralized hub for storing and sharing curated experiment-level datasets, allowing researchers to access a wide range of experimental designs and conditions. This cloud-based infrastructure enhances collaboration and promotes the reproducibility of analyses across different laboratories.

The curatedTCGAData package provides some resources through ExperimentHub, as do many other self-identified "CancerData" resources. Once the ExperimentHub is loaded, it can be queried for terms of interest.

Here and throughout, shading is used to indicate code operations in Bioconductor 3.19 with R 4.4. Lines of output are preceded by ##.

```
library(ExperimentHub)
eh = ExperimentHub()
query(eh, "CancerData")
## ExperimentHub with 1742 records
## # snapshotDate(): 2024-04-29
## # $dataprovider: Eli and Edythe L. Broad Institute
##                of Harvard and MIT, GEO, ...
## # $species: Homo sapiens, Mus musculus, NA
## # $rdataclass: SummarizedExperiment, RaggedExperiment,
##                matrix, list, DFrame,...
## # additional mcols(): taxonomyid, genome, description,
## #   coordinate_1_based, maintainer, rdatadateadded,
##          preparerclass, tags,
## #        rdatapath, sourceurl, sourcetype
## # retrieve records with, e.g., 'object[["EH558"]]'
##
##            title
##   EH558  | ACC_CNASNP-20160128
##   EH559  | ACC_CNVSNP-20160128
##   EH560  | ACC_colData-20160128
##   EH561  | ACC_GISTIC_AllByGene-20160128
##   EH562  | ACC_GISTIC_ThresholdedByGene-20160128
##   ...      ...
##   EH8533 | tcga_transcript_counts
##   EH8534 | target_rhabdoid_wgbs_hg19
##   EH8567 | xenium_hs_breast_addon
##   EH9482 | Capper_example_betas.rda
```

```
##   EH9483 | GIMiCC_Library.rda
```

Multiple terms can be used to narrow results before choosing a download.

```
query(eh, c("CancerData", "esophageal"))
## ExperimentHub with 2 records
## snapshotDate(): 2023-10-24
## $dataprovider: University of California San Francisco
## $species: Homo sapiens
## $rdataclass: RangedSummarizedExperiment, data.frame
## additional mcols(): taxonomyid, genome, description,
##   coordinate_1_based, maintainer, rdatadateadded,
##   preparerclass, tags,}
##   rdatapath, sourceurl, sourcetype }
## retrieve records with, e.g., object[["EH8527"]]
##            title
##   EH8527 | cao_esophageal_wgbs_hg19
##   EH8530 | cao_esophageal_transcript_counts
```

Similarly AnnotationHub files can be downloaded for annotating data. For example, the ensembl 110 release of gene and protein annotations are obtained with the following:

```
library(AnnotationHub)
ah = AnnotationHub()
query(ah, c("ensembl", "110", "Homo sapiens"))
#snapshotDate(): 2024-04-29
#AnnotationHub with 1 record
## snapshotDate(): 2024-04-29
## names(): AH113665
## $dataprovider: Ensembl
## $species: Homo sapiens
## $rdataclass: EnsDb
## $rdatadateadded: 2023-04-25
## $title: Ensembl 110 EnsDb for Homo sapiens
## $description: Gene and protein annotations for Homo
##       sapiens based on Ensem...
## $taxonomyid: 9606
## $genome: GRCh38
## $sourcetype: ensembl
```

```
## $sourceurl: http://www.ensembl.org
## $sourcesize: NA
## $tags: c("110", "Annotation", "AnnotationHubSoftware",
##         "Coverage", "DataImport", "EnsDb", "Ensembl",
##         "Gene", "Protein",
##         "Sequencing", "Transcript")
## retrieve record with 'object[["AH113665"]]'
```

3 Exploring institutionally curated cancer genomics data

3.1 The Cancer Genome Atlas

An overview of Bioconductor's resource for the Cancer Genome Atlas (TCGA) is easy to obtain, with the curatedTCGAData package.

```
library(curatedTCGAData)
tcgatab = curatedTCGAData(version="2.1.1")
```

Records obtained for adrenocortical carcinoma (code ACC) are in Table 1.

Table 1 Records returned by curatedTCGAData::curatedTCGAData(), filtered to those pertaining to adrenocortical carcinoma.

	ah_id	title	file_size	rdataclass
1	EH4737	ACC_CNASNP-20160128	0.8 Mb	RaggedExperiment
2	EH4738	ACC_CNVSNP-20160128	0.2 Mb	RaggedExperiment
3	EH4740	ACC_GISTIC_AllByGene-20160128	0.2 Mb	SummarizedExperiment
4	EH4741	ACC_GISTIC_Peaks-20160128	0 Mb	RangedSummarizedExperiment
5	EH4742	ACC_GISTIC_ThresholdedByGene-20160128	0.2 Mb	SummarizedExperiment
6	EH4744	ACC_Methylation-20160128_assays	239.2 Mb	SummarizedExperiment
7	EH4745	ACC_Methylation-20160128_se	6 Mb	RaggedExperiment
8	EH4747	ACC_Mutation-20160128	0.7 Mb	SummarizedExperiment
9	EH4748	ACC_RNASeq2Gene-20160128	2.7 Mb	SummarizedExperiment
10	EH4750	ACC_RPPAArray-20160128	0.1 Mb	SummarizedExperiment
414	EH8118	ACC_miRNASeqGene-20160128	0.2 Mb	SummarizedExperiment
415	EH8119	ACC_RNASeq2GeneNorm-20160128	5.4 Mb	SummarizedExperiment

Various conventions are in play in this table. The "title" field is of primary concern. The title string can be decomposed into substrings with interpretation `[tumorcode]_[assay]-[date]_[optional codes]`. The column `ah_id` will be explained in section 4, and entries in column `rdataclass` will be discussed in section 6 below.

3.1.1 Tumor code resolution

There are 33 different tumor types available in TCGA. The decoding of tumor codes for the first ten in alphabetical order is provided in Table 2.

3.1.2 Assay codes and counts

Assays performed on tumors vary across tumor types. For assay types shared between breast cancer, glioblastoma, and lung adenocarcinoma (code LUAD), the numbers of tumor and normal samples available in curatedTCGAData are provided in Table 3.

3.1.3 An example dataset for RNA-seq from glioblastoma multiforme

We obtain normalized RNA-seq data on primary tumor samples for GBM with

```
gbrna = TCGAprimaryTumors(curatedTCGAData("GBM",
     "RNASeq2GeneNorm", dry.run=FALSE, version="2.1.1"))
gbrna
## A MultiAssayExperiment object of 1 listed
##experiment with a user-defined name and respective class.
##Containing an ExperimentList class object of length 1:
## [1] GBM_RNASeq2GeneNorm-20160128: SummarizedExperiment
##        with 18199 rows and 153 columns
##
## Functionality:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DataFrame
##  sampleMap() - the sample coordination DataFrame
##  `$`, `[`, `[[` - extract colData columns, subset, or
##       experiment
## *Format() - convert into a long or wide DataFrame
## assays() - convert ExperimentList to a SimpleList of
##       matrices
## exportClass() - save data to flat files
```

Table 2 Decoding TCGA tumor code abbreviations.

Code	Tumor.Type
ACC	Adrenocortical Carcinoma
BLCA	Bladder Urothelial Carcinoma
BRCA	Breast Invasive Carcinoma
CESC	Cervical Squamous Cell Carcinoma and Endocervical Adenocarcinoma
CHOL	Cholangiocarcinoma
CNTL	Controls
COAD	Colon Adenocarcinoma
DLBC	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma
ESCA	Esophageal Carcinoma
FPPP	FFPE Pilot Phase II
GBM	Glioblastoma Multiforme
HNSC	Head and Neck Squamous Cell Carcinoma
KICH	Kidney Chromophobe
KIRC	Kidney Renal Clear Cell Carcinoma
KIRP	Kidney Renal Papillary Cell Carcinoma
LAML	Acute Myeloid Leukemia
LCML	Chronic Myelogenous Leukemia
LGG	Brain Lower Grade Glioma
LIHC	Liver Hepatocellular Carcinoma
LUAD	Lung Adenocarcinoma
LUSC	Lung Squamous Cell Carcinoma
MESO	Mesothelioma
MISC	Miscellaneous
OV	Ovarian Serous Cystadenocarcinoma
PAAD	Pancreatic Adenocarcinoma
PCPG	Pheochromocytoma and Paraganglioma
PRAD	Prostate Adenocarcinoma
READ	Rectum Adenocarcinoma
SARC	Sarcoma
SKCM	Skin Cutaneous Melanoma
STAD	Stomach Adenocarcinoma
TGCT	Testicular Germ Cell Tumors
THCA	Thyroid Carcinoma
THYM	Thymoma
UCEC	Uterine Corpus Endometrial Carcinoma
UCS	Uterine Carcinosarcoma
UVM	Uveal Melanoma

Table 3 Numbers of assays available in TCGA on tumor and normal samples, for breast cancer, glioblastoma, and lung adenocarcinoma.

	BRCA	BRCAnormal	GBM	GBMnormal	LUAD	LUADnormal
CNASNP	1089	1120	577	527	516	579
CNVSNP	1080	1119	577	527	516	579
GISTIC_AllByGene	1080	0	577	0	516	0
GISTIC_Peaks	1080	0	577	0	516	0
GISTIC_ThresholdedByGene	1080	0	577	0	516	0
Mutation	988	5	283	7	230	0
RNASeq2Gene	1093	119	153	13	515	61
RPPAArray	887	50	233	11	365	0
RNASeq2GeneNorm	1093	119	153	13	515	61
Methylation_methyl27	314	29	285	0	65	24
Methylation_methyl450	783	102	140	14	458	34

R functions defined in Bioconductor packages can operate on the variable `gbrna` to retrieve information of interest. Details on the underlying data structure are given in section 6 below. For most assay types, we think of the quantitative assay information as tabular in nature, with table rows corresponding to genomic features such as genes, and table columns corresponding to samples.

Information on GBM samples employs the `colData` function.

```
dim(colData(gbrna))
## [1] 153 4380
```

For sample level information obtained `colData`, we think of rows as samples, and columns as sample attributes.

3.1.4 Clinical and phenotypic data

TCGA datasets are generally provided as combinations of results for tumor tissue and normal tissue. The determination of a record's sample type is encoded in the sample "barcode". Decoding of sample barcodes is described at

`https://docs.gdc.cancer.gov/Encyclopedia/pages/TCGA_Barcode/`

with specific interpretation of sample types listed at

`https://gdc.cancer.gov/resources-tcga-users/tcga-code-tables/sample-type-codes`

separately. The TCGAutils package provides utilities for extracting data on primary tumor samples, excluding samples that may have been taken on normal tissue or metastases.

Clinical and phenotypic data on all TCGA samples are voluminous. For example, there are 2684 fields of sample level data for BRCA samples, and 4380 fields for GBM samples. Many of these fields are meaningfully populated for only a very small minority of samples. To see this for GBM:

```
mean(sapply(colData(gbrna), function(x) mean(is.na(x))>.90))
## [1] 0.8091324
```

In words, for 81% of clinical data fields in TCGA GBM data, at least 90% of entries are missing.

Nevertheless, with careful inspection of fields and contents, nearly complete clinical data can be extracted and combined with molecular and genetic assay data with modest effort.

The following code chunk illustrates a very crude approach to comparing survival profiles for BRCA, GBM, and LUAD donors. The result is in Figure 3.

```
# obtain mutation data for BRCA, GBM, LUAD; could use any or
#    all assay types
brmut = curatedTCGAData("BRCA", "Mutation", version = "2.1.1",
      dry.run = FALSE)
gbmut = curatedTCGAData("GBM", "Mutation", version = "2.1.1",
      dry.run = FALSE)
lumut = curatedTCGAData("LUAD", "Mutation", version = "2.1.1",
      dry.run = FALSE)
# extract survival times
library(survival)
getSurv = function(mae) {
 days_on = with(colData(mae), ifelse(is.na(days_to_last_followup),
 days_to_death, days_to_last_followup))
 Surv(days_on, colData(mae)$vital_status)
}
ss = lapply(list(brmut, gbmut, lumut), getSurv)
codes = c("BRCA", "GBM", "LUAD")
type = factor(rep(codes, sapply(ss,length)))
allsurv = do.call(c, ss)
library(GGally)
ggsurv(survfit(allsurv~type))
```

At this point, survival times within tumor type can be stratified by any features of the mutation profiles of individual samples. The "RaggedExperiment" class is employed to test each BRCA sample for presence of any mutation in the gene TTN. See Figure 4.

Fig. 3 Survival profile extraction from three MultiAssayExperiments produced with curatedTCGAData calls.

```
bprim = TCGAprimaryTumors(brmut)
## harmonizing input:
## removing 5 sampleMap rows with 'colname' not in
##      colnames of experiments
mutsyms = assay(experiments(bprim)[[1]], "Hugo_Symbol")
cn = rownames(colData(bprim)) # short
cna = colnames(mutsyms) # long
cnas = substr(cna, 1, 12)
hasTTNmut = apply(assay(experiments(
    TCGAprimaryTumors(brmut))[[1]],
     "Hugo_Symbol"), 2,
     function(x) length(which(x=="TTN"))>0)
## harmonizing input:
## removing 5 sampleMap rows with 'colname' not in
##      colnames of experiments
names(hasTTNmut) = cnas
bsurv = getSurv(TCGAprimaryTumors(brmut))
## harmonizing input:
## removing 5 sampleMap rows with 'colname' not in
##      colnames of experiments
hasTTNmut = hasTTNmut[cn] # match mut records to surv times
ggsurv(survfit(bsurv~hasTTNmut))
```

Similar manipulations permit exploration of relationships between any molecular assay outcomes and any clinical data collected in TCGA.

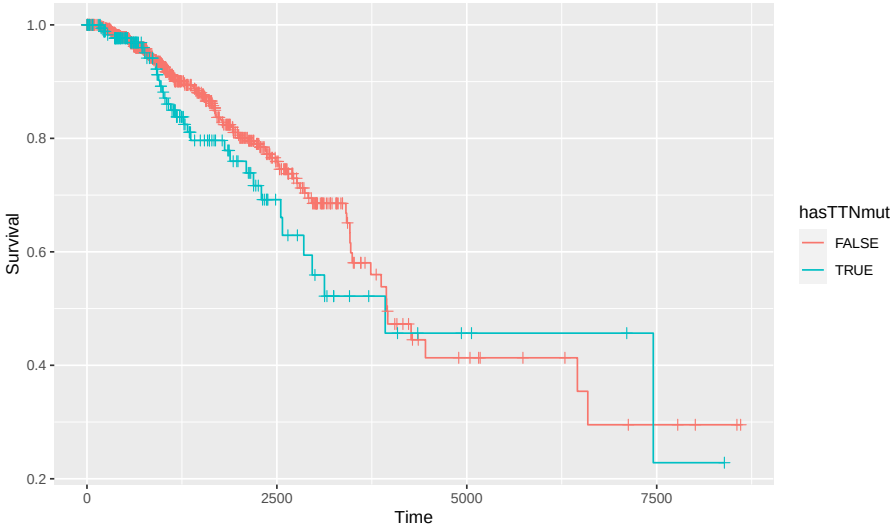


Fig. 4 Survival distributions for donors of breast tumors in TCGA, stratified by presence or absence of mutation in gene TTN.

3.2 cBioPortal

The cBioPortal user guide at

`https://www.cbioportal.org/`

defines the goal of the portal to be reducing "the barriers between complex genomic data and cancer researchers by providing rapid, intuitive, and high-quality access to molecular profiles and clinical attributes from large-scale cancer genomics projects, and therefore to empower researchers to translate these rich data sets into biologic insights and clinical applications."

Bioconductor's cBioPortalData package simplifies access to over 300 genomic studies of diverse cancers in cBioPortal. The main unit of data access is the publication. The `cBioPortal` function mediates a connection between an R session and the cBioPortal API. `getStudies` returns a tibble with metadata on all studies.

```
library(cBioPortalData)
cbio = cBioPortal()
allst = getStudies(cbio)
dim(allst)
## [1] 397 13
```

A pruned selection of records from the cBioPortal studies table is given in Table 4.

Table 4 Excerpts from four fields on selected records in the cBioPortal getStudies output.

name	description	studyId
Adenoid Cystic Carcinoma of the Breast	Whole exome sequencing of 12 breast AdCCs.	acbc_mskcc_2015
Adenoid Cystic Carcinoma	Whole-exome or whole-genome sequencing analysis of 60 ACC tumor/normal pairs	acyc_mskcc_2013
Adenoid Cystic Carcinoma	Targeted Sequencing of 28 metastatic Adenoid Cystic Carcinoma samples.	acyc_fmi_2014
Adenoid Cystic Carcinoma	Whole-genome or whole-exome sequencing of 25 adenoid cystic carcinoma tumor/normal pairs.	acyc_jhu_2016
Adenoid Cystic Carcinoma	WGS of 21 salivary ACCs and targeted molecular analyses of a validation set (81 patients).	acyc_mda_2015
Adenoid Cystic Carcinoma	Whole-genome/exome sequencing of 10 ACC PDX models.	acyc_mgh_2016
Adenoid Cystic Carcinoma	Whole exome sequencing of 24 ACCs.	acyc_sanger_2013
Adenoid Cystic Carcinoma Project	Multi-Institute Cohort of 1045 Adenoid Cystic Carcinoma patients.	acc_2019
Basal Cell Carcinoma	Whole-exome sequencing of 126 basal cell carcinoma tumor/normal pairs; targeted sequencing of 163 sporadic samples (40 tumor/normal pairs) and 4 Gorlin syndrome basal cell carcinomas.	bcc_unige_2016

To explore copy number alteration data from a study on angiosarcoma, we find the associated studyId field in `allst` and use the `cBioDataPack` function to retrieve a MultiAssayExperiment:

```
ann = "angs_project_painter_2018"
ang = cBioDataPack(ann)
ang
## A MultiAssayExperiment object of 3 listed
##experiments with user-defined names and respective classes.
####Containing an ExperimentList class object of length 3:
##[1] cna_hg19.seg: RaggedExperiment with 27835 rows
##           and 48 columns
##[2] cna: SummarizedExperiment with 23109 rows
##           and 48 columns
##[3] mutations: RaggedExperiment with 24058 rows
##           and 48 columns
## Functionality:
##  experiments() - obtain the ExperimentList instance
##  colData() - the primary/phenotype DataFrame
##  sampleMap() - the sample coordination DataFrame
##  '$', '[', '[[' - extract colData columns, subset, or
```

```
##      experiment
## *Format() - convert into a long or wide DataFrame
## assays() - convert ExperimentList to a SimpleList of
##      matrices
## exportClass() - save data to flat files
```

The copy number alteration outcomes are in the `assay` component of the experiment.

```
seg = experiments(ang)[[1]]
colnames(seg) = sapply(strsplit(colnames(seg), "-"), "[", 5)
assay(seg)[1:4,1:4]
##
##                   DAE1F DACME DADBW DAD34
## 1:12227-955755       71    NA    NA    NA
## 1:957844-1139868     62    NA    NA    NA
## 1:1140874-1471177   167    NA    NA    NA
## 1:1475170-1855370   113    NA    NA    NA
```

The rownames component of this matrix can be transformed to a GenomicRanges instance for concise manipulation.

```
allalt = GRanges(rownames(assay(seg)))
 allalt
## GRanges object with 27835 ranges and 0 metadata columns:
##           seqnames            ranges strand
##              <Rle>         <IRanges>  <Rle>
##       [1]        1      12227-955755      *
##       [2]        1    957844-1139868      *
##       [3]        1   1140874-1471177      *
##       [4]        1   1475170-1855370      *
##       [5]        1  1857786-17257894      *
##       ...      ...               ...    ...
##   [27831]       20     68410-1559342      *
##   [27832]       20   1585705-1592359      *
##   [27833]       20  1616247-62904955      *
##   [27834]       21  9907492-48084286      *
##   [27835]       22 16157938-51237572      *
##   -------
##   seqinfo: 22 sequences from an unspecified genome; no
```

```
##        seqlengths
```

We'll focus on chromosome 17, where TP53 is found. Regions of genomic alteration are summarized to their midpoints. The display in Figure 5 shows a strong peak in the vicinity of 7.5 Mb on chromosome 17, near TP53.

```
g17 = allalt[seqnames(allalt)=="17"]
df17 = as(g17, "data.frame")
df17$mid = .5*(df17$start+df17$end) # midpoint only
ggplot(df17, aes(x=mid)) + geom_density(bw=.2) +
   xlab("chr 17 bp")
```

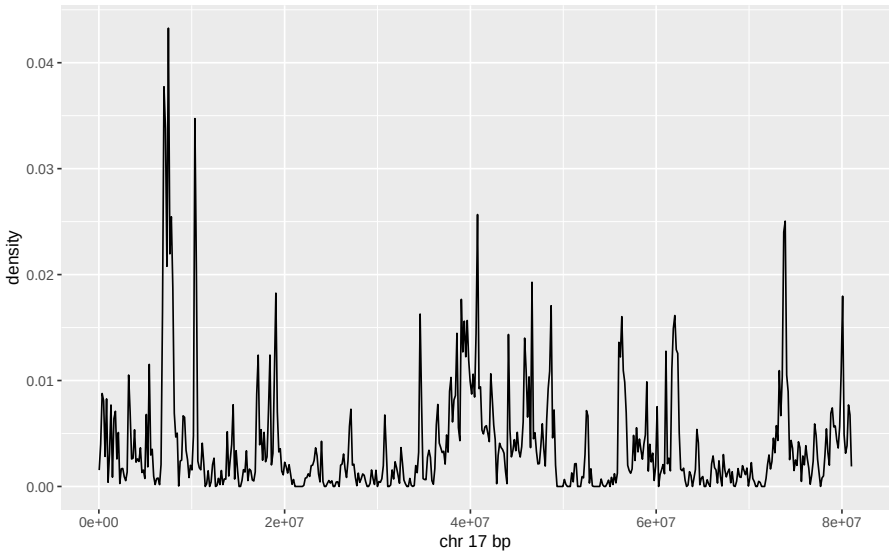


Fig. 5 Density of recurrent genomic alterations on chromosome 17 for 48 angiosarcoma patients.

4 Genomic annotation resources relevant to cancer

4.1 Resources from UCSC, NCBI, and EMBL

Sequences for reference genome builds for human and other model organisms are supplied in BSgenome packages. BSgenome.Hsapiens.UCSC.hg19 provides all chromosomes and contigs for the 2009 build; the hg38 suffix may be used for the 2013 build. The recent "telomere to telomere" build is available as BSgenome.Hsapiens.NCBI.T2T.CHMv13v2.0.

NCBI's dbSNP catalog of genetic variants is provided in versioned packages. For example, SNPlocs.Hsapiens.dbSNP155.GRCh38 includes position and nucleotide content information for over 1 billion SNP identifiers ("rs numbers").

Tracks defined for the UCSC genome browser are also packaged. The package

`TxDb.Hsapiens.UCSC.knownGene.hg38`

can be used to get gene, transcript, and exon location information for the hg38 build. The EnsDb packages provide similar information for annotations curated at EMBL.

```
library(EnsDb.Hsapiens.v86)
EnsDb.Hsapiens.v86
## EnsDb for Ensembl:
## |Backend: SQLite
## |Db type: EnsDb
## |Type of Gene ID: Ensembl Gene ID
## |Supporting package: ensembldb
## |Db created by: ensembldb package from Bioconductor
## |script\_version: 0.3.0
## |Creation time: Thu May 18 16:32:27 2017
## |ensembl\_version: 86
## |ensembl\_host: localhost
## |Organism: homo\_sapiens
## |taxonomy\_id: 9606
## |genome\_build: GRCh38
## |DBSCHEMAVERSION: 2.0
## | No. of genes: 63970.
## | No. of transcripts: 216741.
## |Protein data available.
```

The "genes" method provides addresses and additional annotations.

```
names(mcols(genes(EnsDb.Hsapiens.v86)))
```

```
## [1] "gene_id"          "gene_name"        "gene_biotype"
## [4] "seq_coord_system" "symbol"           "entrezid"
head(table(genes(EnsDb.Hsapiens.v86)$gene_biotype))
##      3prime_overlapping_ncRNA                      antisense
##                            30                           5703
## bidirectional_promoter_lncRNA                      IG_C_gene
##                             4                             23
##               IG_C_pseudogene                      IG_D_gene
##                            11                             64
```

More recent versions of Ensembl gene annotation are available from AnnotationHub, as illustrated above in section 2.5 with the creation of `ens110`.

4.2 Gene sets

Many methods have been developed to employ collections of genes for inference on hypotheses about cancer initiation or progression. The Molecular Signatures Database (MSigDB) is curated at Broad Institute, and can be harvested using the msigdb package.

Collect all gene sets for humans:

```
library(msigdb)
hssigs = getMsigdb(org="hs", id="SYM",
   version=getMsigdbVersions())
```

Find those with CANCER in their name:

```
nms = grep("CANCER", names(hssigs), value=TRUE)
head(nms)
## [1] "SOGA_COLORECTAL_CANCER_MYC_DN"
## [2] "SOGA_COLORECTAL_CANCER_MYC_UP"
## [3] "WATANABE_RECTAL_CANCER_RADIOTHERAPY_RESPONSIVE_UP"
## [4] "LIU_PROSTATE_CANCER_UP"
## [5] "BERTUCCI_MEDULLARY_VS_DUCTAL_BREAST_CANCER_UP"
## [6] "WATANABE_COLON_CANCER_MSI_VS_MSS_UP"
wangmet = hssigs[["WANG_METASTASIS_OF_BREAST_CANCER_ESR1_UP"]]
wangmet
## setName: WANG_METASTASIS_OF_BREAST_CANCER_ESR1_UP
```

```
## geneIds: KPNA2, HDGFL3, ..., PSMC2 (total: 22)
## geneIdType: Symbol
## collectionType: Broad
##bcCategory: c2 (Curated)
##bcSubCategory: CGP
## details: use 'details(object)'
```

Information on provenance is bound together with the gene list:

```
details(wangmet)
## setName: WANG_METASTASIS_OF_BREAST_CANCER_ESR1_UP
## geneIds: KPNA2, HDGFL3, ..., PSMC2 (total: 22)
## geneIdType: Symbol
## collectionType: Broad
##bcCategory: c2 (Curated)
##bcSubCategory: CGP
## setIdentifier: LVY1HGGWMJ7:35020:Fri May 26 13:33:02
##          2023:1104005
## description: Genes whose expression in primary ER(+)
##      [GeneID=2099] breast cancer tumors positively correla
## (longDescription available)
## organism: Homo sapiens
## pubMedIds: 15721472
## urls: https://data.broadinstitute.org/gsea-msigdb/msigdb/
##    release/2023.1.Hs/msigdb_v2023.1.Hs.xml.zip
## contributor: Arthur Liberzon
```

4.3 Ontologies

Informal reasoning about cancer genomics employs conventional but frequently ambiguous terminology. In modern information science, ontologies are structured vocabularies (sets of "terms", which may be single words or natural language phrases) accompanied by explicit statements of semantic relationships among terms.

Bioconductor provides several approaches for using ontologies in cancer data science. The most familiar ontology in this domain is Gene Ontology (GO), which organizes vocabulary about genes and gene products in the areas of molecular function, cellular components, and biological processes.

4.3.1 Ontology usage with AnnotationDbi

A common use case is to find genes or proteins associated with some biological process, component, or function. A phrase like 'Golgi membrane' can be found in Gene Ontology using the select method with GO.db:

```
library(GO.db)
select(GO.db, keytype="TERM",
keys="Golgi membrane", columns=c("GOID", "DEFINITION",
       "ONTOLOGY"))
##             TERM       GOID
## 1 Golgi membrane GO:0000139
##                                  DEFINITION
## 1 The lipid bilayer surrounding any of the
## compartments of the Golgi apparatus.
## ONTOLOGY
## 1     CC
```

Once the formal identifier is obtained, the org.Hs.eg.db package can be used to find mappings from the GO term to gene and protein identifiers. This generates a fairly large table:

```
library(org.Hs.eg.db)
go139 = select(org.Hs.eg.db, keys="GO:0000139", keytype="GO",
columns=c("ENTREZID", "SYMBOL", "PFAM"))
dim(go139)
## [1] 1212 6
head(go139)
##           GO EVIDENCE ONTOLOGY ENTREZID SYMBOL    PFAM
## 1 GO:0000139      TAS       CC       28    ABO PF03414
## 2 GO:0000139      IEA       CC      102 ADAM10 PF00200
## 3 GO:0000139      IEA       CC      102 ADAM10 PF13574
## 4 GO:0000139      IEA       CC      102 ADAM10 PF01562
## 5 GO:0000139      TAS       CC      162  AP1B1 PF09066
## 6 GO:0000139      TAS       CC      162  AP1B1 PF01602
```

The evidence code TAS means that there is a "traceable author statement" associating the term of interest with the gene identified. The number of genes in traceable Golgi membrane:gene associations is found with

```
go139 |> dplyr::filter(EVIDENCE=="TAS") |>
    distinct(ENTREZID) |> count()
##       n
## [1] 327
```

4.3.2 Ontology usage with rols

Access to a vast collection of ontologies is afforded by the EBI's Ontology Lookup Service (OLS). The rols package uses the OLS API to discover ontologic mapping of terms of interest. Here we'll consider the term "golgi membrane dynamics", which is not found in GO. Again a multistep process is used.

```
library(rols)
lk1 = OlsSearch(q="golgi membrane dynamics", exact=TRUE)
lk1
## Object of class 'OlsSearch':
##query: golgi membrane dynamics
##requested: 20 (out of 3)
##response(s): 0
```

In this first step, we find how extensive is the response to the query. Certain searches yield tens of thousands of hits. With the exact parameter setting, the yield is modest. Now we extract a data.frame after requesting all records with `olsSearch`. Results are excerpted in Table 5.

```
lk2 = olsSearch(lk1)
lk3 = as(lk2, "data.frame")
lk3$description = unlist(lk3$description)
```

The detailed descriptions of the NCI Thesaurus entries show the exact nature of the search outcome.

4.3.3 Cross-ontology relationships

Philosophically, ontology is the study of what there is. For applications in information science, boundaries need to be established so that ontological resources can be

Table 5 Using rols to obtain ontologic information related to golgi membrane dynamics.

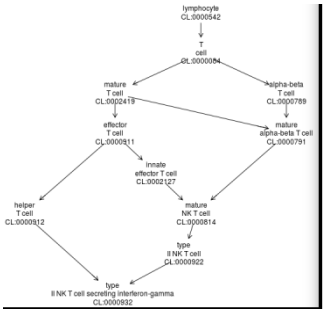
short_form	description	label
NCIT_C119637	This gene is involved in both protein ubiquitination and Golgi membrane dynamics.	HACE1 Gene
NCIT_C119639	E3 ubiquitin-protein ligase HACE1 (909 aa, ~102 kDa) is encoded by the human HACE1 gene. This protein is involved in the regulation of both the ubiquitination and subsequent degradation of small GTPases, which modulates Golgi membrane dynamics.	E3 Ubiquitin-Protein Ligase HACE1
NCIT_C119638	Human HACE1 wild-type allele is located in the vicinity of 6q16.3 and is approximately 132 kb in length. This allele, which encodes E3 ubiquitin-protein ligase HACE1 protein, plays a role in the modulation of both Golgi membrane dynamics and ubiquitination. Mutations of the gene, including translocations that either reduce expression of the gene (t(6;15)(q21;q21)) or truncate the gene (t(5;6)(q21;q21)), are associated with Wilms tumor.	HACE1 wt Allele

managed with well-defined scopes. In Gene Ontology, three sub-ontologies are explicitly identified for cellular components, biological processes, and molecular functions.

As knowledge of cell biology increases, the typology of cells becomes more and more intricate. Differentiation and definition of "cell types" involves concepts from immunology, protein science, anatomy, and other conceptual domains for which ontologies have been developed. Figure 6 presents, on the left, the hierarchy of cell type concepts starting at "lymphocyte", leading to "Type II Natural Killer T cell secreting interferon gamma". On the right, some of the GO and Protein Ontology (PR) cross-references in the Cell Ontology (CL) entry for the Type II NK cell are shown. The "cond" column of the table contains abbreviated tokens representing formal relationships linking the cell type to the protein or cellular component elements of PR and GO. The token "hasPMP" refers to the element of the Relation Ontology (RO) "has plasma membrane part" (RO:0002104).

Prospects for use of ontological discipline in the definition of new cell types are reviewed in a 2018 paper from the Venter Institute [2].

The field of biological ontology is rapidly advancing, and the integration of ontology search and inference with data analytic frameworks requires more effort at this time.



query	tag	prtag	cond	entity	SYMBOL	name
CL:0000932	CL:0000932	PR:000001343	hasPMP	CD69 molecule	CD69	type II NK T cell secreting interferon-gamma
CL:0000932	CL:0000814	PR:000001874	hasPMP	KLRB1-like protein		mature NK T cell
CL:0000932	CL:0000789	GO:0042105	hasPMP	alpha-beta T cell receptor complex		alpha-beta T cell
CL:0000932	CL:0000542	CP:0000000	hasPart			lymphocyte
CL:0000932	CL:0000542	CP:0000037	hasPart			lymphocyte
CL:0000932	CL:0000542	GO:0000792	hasPart	heterochromatin		lymphocyte
CL:0000932	CL:0000542	GO:0030141	lacksPart	secretory granule		lymphocyte
CL:0000932	CL:0002419	GO:0042101	hasPMP	T cell receptor complex		mature T cell

Fig. 6 Ontology visualization and tabulation with ontoProc::ctmarks.

5 Analytical workflows

5.1 Overview

Table 6 presents an informal topical labeling for Bioconductor software packages with cancer mentioned in the Description field of package metadata.

The vignettes of each of these packages provide background and illustration of their roles in cancer genomics.

5.2 Packages supporting epigenomic analysis

Bioconductor also provides a diverse array of packages for analysis of epigenome data. Cancer is often studied under a developmental lens, so increasingly, studies are measuring cell states using epigenomic methods. Epigenomics is the study of chemical modifications and chromosomal conformations of DNA in a nucleus; in cancer epigenomics, we study how the cancer epigenome differs among cancers and how these relate to healthy epigenomes. As of 2023, Bioconductor includes 89 packages under *Epigenetics* and 93 packages tagged under *FunctionalGenomics*, including dozens of tools for analyzing a variety of epigenome assays, such as ATAC-seq, ChIP-seq, or bisulfite-seq. Among these are also tools that handle more general analysis, such as genomic region set enrichment.

First, for ATAC-seq data, bioconductor packages include general-purpose pipelines, including scPipe [3]. and esATAC [4] which start from FASTQ files and produce feature count matrices. Alternatively, many practitioners elect to do general-purpose pipeline processing outside of R, and then bring the processed data into R for statistical analysis, visualization, and quality control. In this approach, ATACseqQC provides a variety of QC plots specific to ATAC-seq data [5].

For DNA methylation, many popular packages have been developed to help with all stages of a DNA methylation analysis. These include minfi [6] which specializes in

Table 6 Topical organization of packages with cancer applications.

topic	packages
Ancestry	RAIDS
Biomarkers	INDEED, iPath, RLassoCox
ceRNA	GDCRNATools
Clonal Evolution	CIMICE, LACE, OncoSimulR, TRONCO, CancerInSilico, cellscape
CNV	oncoscanR, SCOPE, ZygosityPredictor
DrugSensitivity	DepInfeR, octad, PharmacoGx, rcellminer
Epigenetics	MethylMix, AMARETTO, COCOA, methylclock, missMethyl
HotSpots/Drivers/signatures	compSPOT, MoonlightR, Moonlight2R, DriverNet, genefu, mastR, pathifier, RESOLVE, macat, SigCheck, signeR, signifinder, supersigs, decompTumor2Sig, YAPSA
ImmuneModulation	easier
IsoformSwitching	IsoformSwitchAnalyzeR
Literature mining	OncoScore
ncRNA	NoRCE
Radiomics	RadioGx
RecurrentFusion	copa, oppar
Spatial	SpatialDecon
SpecificCancers	consensusOV, PDATK, STROMA4
Splicing	OutSplice, psichomics
Subtyping	SCFA

methylation array analysis, biseq and bsseq [7] which provide fundamental infrastructure for sequencing-based assays, and RnBeads [8], which provides a comprehensive general-purpose analysis of DNA methylation cohorts from arrays or sequencing-based assays. Other packages provide more specialized analysis approaches, such as MIRA [9], which infers regulatory activity of transcription factors using DNA methylation signals, or ELMER, which uses DNA methylation and gene expression in large cancer cohorts to infer transcription factor networks [10]. EpiDISH infers the proportions of cell-types present in a bulk sample on the basis of DNA methylation data [11].

DiffBind [12] facilitates differential binding analysis of ChIP-seq peak data.

GenomicDistributions [13] provides a variety of plots for visualization distributions of any type of genomic range data. The chromPlot package specializes in plots across chromosomes. Several packages deal with unsupervised exploration of variation in epigenomic data. PathwayPCA, MOFA2 [14] and COCOA [15] can process any epigenomic signal data. A variety of alternative approaches for enrichment analysis, which include LOLA [16], chipenrich, regionR [17], and FGNet [18]. Annotation packages include ChIPpeakAnno [19] and annotatr [20].

5.3 Some details on prediction of responsiveness to immune checkpoint blockade

The National Cancer Institute website on checkpoint inhibitors in cancer immunotherapy ("Immune Checkpoint Inhibitors" [21]) lists 12 different cancer types amenable to treatment via immune checkpoint inhibition. The "easier" package in Bioconductor assembles multiple systems biology resources to produce patient-specific prediction of responsiveness to immune checkpoint blockade (ICB) [22].

Figure 7 presents on overview of results of immune response assessment in a cohort of patients with bladder cancer [23]. Patient's bulk RNA-seq data are used to develop multiple quantitative descriptors of the tumor microenvironment, and scores for processes regarded as hallmarks of anti-cancer immune responses.

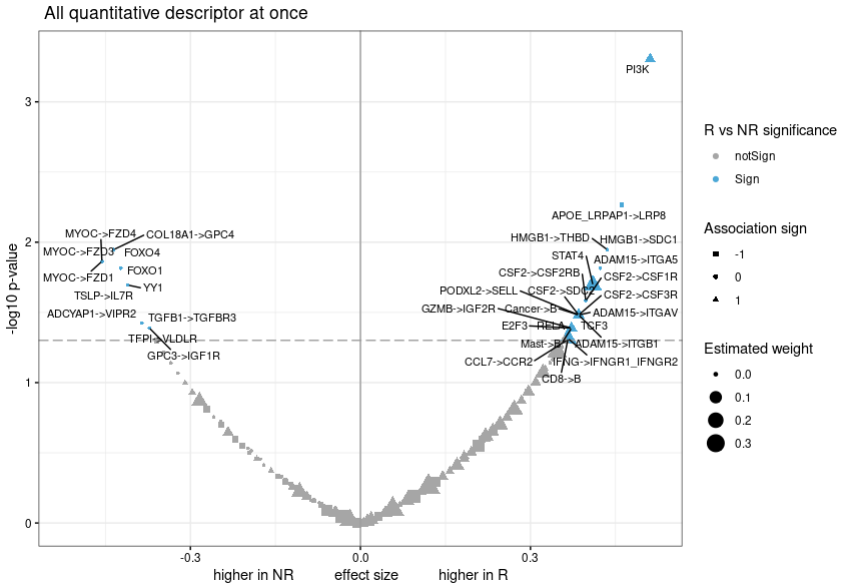


Fig. 7 Comparison of genomic features distinguishing patients non-responsive and responsive to immune checkpoint blockade.

This display encapsulates a) the capacity of measurements of genomic elements to discriminate patients who respond to ICB for bladder cancer (position of labeled item on x axis), b) the direction of association of element activity with immune response (shape of glyph) and c) the relative magnitudes of weights (size of glyph) estimated for features in initial model fitting.

The design of this package is noteworthy in its approach to information hiding. Parameters estimated in machine learning of tissue-specific relations between quantitative descriptors of the tumor microenvironment and hallmarks of immune response are stored in ExperimentHub.

```
library(easierData)
list_easierData()
##   eh_id                            title
##  EH6677   Mariathasan2018_PDL1_treatment
##  EH6678                        opt_models
##  EH6679                 opt_xtrain_stats
##  EH6680              TCGA_mean_pancancer
##  EH6681                TCGA_sd_pancancer
##  EH6682                 cor_scores_genes
##  EH6683               intercell networks
##  EH6684                lr_frequency_TCGA
##  EH6685                   group_lr_pairs
##  EH6686                  HGNC_annotation
##  EH6687           scores_signature_genes
```

The structure of the stored model weights resource can be sketched by probing list elements.

```
mw = eh[["EH6678"]]
## see ?easierData and browseVignettes('easierData') for
##     documentation
## loading from cache
names(mw)   # TCGA tumor types
##  [1] "LUAD" "LUSC" "BLCA" "BRCA" "CESC" "CRC"  "GBM"
##         "HNSC" "KIRC"
## [10] "KIRP" "LIHC"   "OV" "PAAD" "PRAD" "SKCM" "STAD"
##         "THCA" "UCEC"
## [19] "NSCLC"
names(mw[["LUAD"]]) # TME descriptors
## [1] "pathways" "immunecells" "tfs" "lrpairs" "ccpairs"
rownames(mw[["LUAD"]]$pathways$CYT) # predict cytolytic
#                                   # activity
##  [1] "(Intercept)" "Androgen" "EGFR" "Estrogen" "Hypoxia"
##  [6]    "JAK-STAT"     "MAPK" "NFkB" "p53"      "PI3K"
## [11]        "TNFa"    "Trail" "VEGF" "WNT"
```

The vignette of the easier package steps through phases, using these tumor-type-specific weights to compute patient-specific measures of transcription factor activity or cell-cell interaction on the basis of bulk RNA-seq (units are transcripts per million), and a patient-specific measure of pathway activity using raw RNA-seq

counts. These metrics may be of interest in their own right for applications other than establishing predictions of response to ICB.

Section 9 provides the names and versions of all packages used to produce this analysis.

5.4 Representing and visualizing spatial transcriptomics experiments

Spatial transcriptomics (ST) allows the quantification of RNA expression of large numbers of genes while preserving the spatial context of tissues and cells. This is important as cancer progression depends on a complex tumor microenvironment, and not just cell type composition, but also cell type spatial organization can be used to derive diagnostic or prognostic markers.

The Bioconductor project offers multiple approaches to handle and manipulate spatial transcriptomics data. The SpatialExperiment class [24] is designed to be a lightweight, technology-agnostic container. By inheriting from the SingleCellExperiment class, it unlocks the use in ST data of analysis packages designed for single-cell data, such as scater for exploration and QC, and scran for normalization. SpatialFeatureExperiment [25] extends SpatialExperiment to easily reuse polygons and other spatial geometry features from geospatial CRAN packages, such as sf. See also MoleculeExperiment [26] for a different approach based on the data.table package.

In addition to data containers, Bioconductor provides a rich set of ST data. The STexampleData and SFEData packages contain a collection of datasets from different technologies and tissues. As of December 2023, the TENxVisiumData package provides a collection of 13 in-house 10X Genomics Visium datasets from 23 samples across two organisms (human and mouse) and 13 tissues. The MerfishData package contains two annotated samples assayed with the MERFISH in-situ imaging protocol.

Finally, Bioconductor offers a growing collection of analysis methods tailored for spot-based and in-situ ST data, including methods for visualization, data exploration and quality control, spot deconvolution, spatially-aware clustering, and identification of spatially-variable genes.

To show a simple example of an analysis workflow on spot-based data, we explore a fresh frozen Invasive Ductal Carcinoma breast tissue assayed with the 10X Genomics Visium platform. First, we use the ggspavis package for visualization. See Figure 8.

```
library(TENxVisiumData)
## snapshotDate(): 2023-10-24
library(SpatialExperiment)
library(ggspavis)
hbc <- HumanBreastCancerIDC()
```

```
## see ?TENxVisiumData and browseVignettes('TENxVisiumData')
##        for documentation
## loading from cache
hbc <- hbc[,hbc$sample_id=="HumanBreastCancerIDC1"]
hbc$in_tissue <- TRUE
hbc <- rotateImg(hbc, degrees=-90)
plotVisium(hbc, y_reverse = FALSE)
```

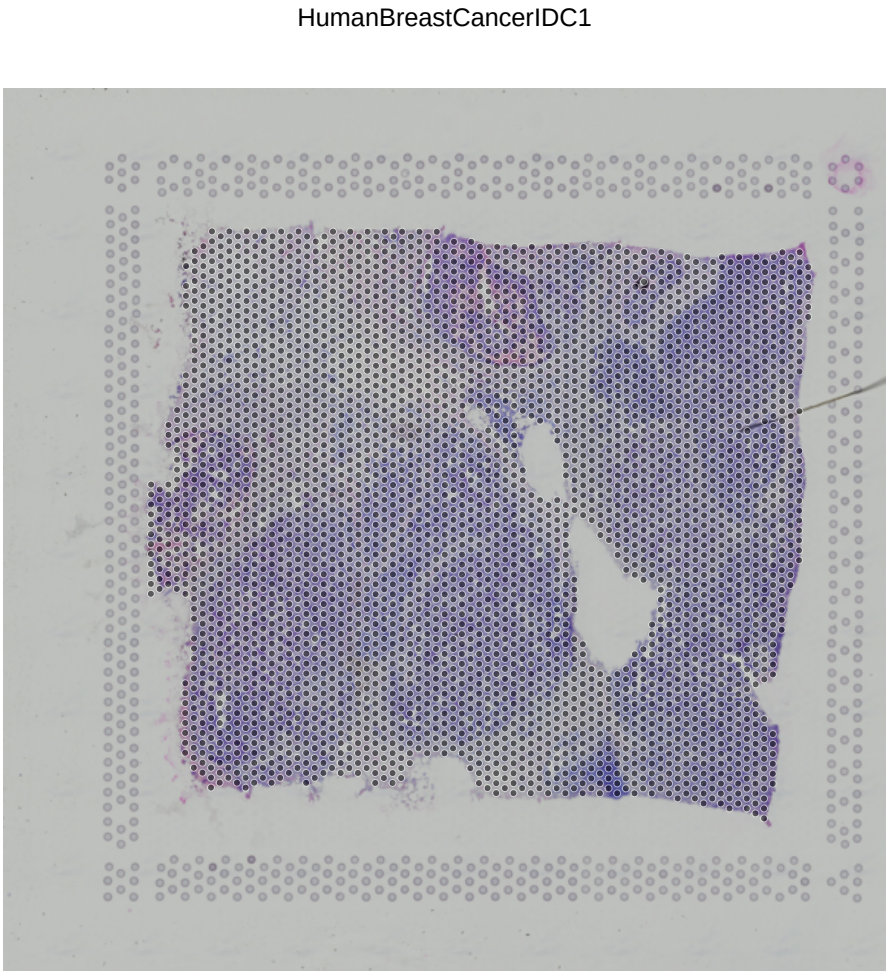


Fig. 8 Visualization of a Visium breast cancer sample

To investigate the spatially variable genes the nnSVG package implements a method for the detection of genes whose expression varies in the tissue spatial domains by fitting nearest-neighbor Gaussian processes [27].

```
library(scater)
library(nnSVG)
library(scran)
#add quality metrics
is_mito <- grepl("(^MT-)|(^mt-)", rowData(hbc)$symbol)
hbc <- addPerCellQC(hbc, subsets = list(mito = is_mito))
## needed because the column name is hard coded in
##      the nnSVG::filter_genes
rowData(hbc)$gene_name <- rowData(hbc)$symbol
## filter and normalize gene expression
hbc <- filter_genes(hbc)
## Gene filtering: removing mitochondrial genes
## removed 13 mitochondrial genes
## Gene filtering: retaining genes with at least 3 counts
##       in at least 0.5% (n = 19) of spatial locations
## removed 26583 out of 36588 genes due to low expression
hbc <- computeLibraryFactors(hbc)
hbc <- logNormCounts(hbc)
## select highly variable genes
hvgs <- getTopHVGs(hbc, n=1000)
hbc <- hbc[hvgs,]
## identify spatially variable genes
hbc <- nnSVG(hbc, n_threads=4)
## post-processing
hbc <- hbc[order(rowData(hbc)$rank),]
gnr1 <- rowData(hbc)$symbol[1]
rownames(hbc) <- rowData(hbc)$symbol
```

By ranking the results of nnSVG, we are able to detect the most spatially variable genes. As an example, we show how the most spatially variable gene varies across the tissue.

```
plotVisium(hbc, y_reverse = FALSE, fill = gnr1, palette="red")
```

Finally, we show an example of an in-situ ST technology, by visualizing a breast cancer sample assayed with the 10X Genomics Xenium platform.

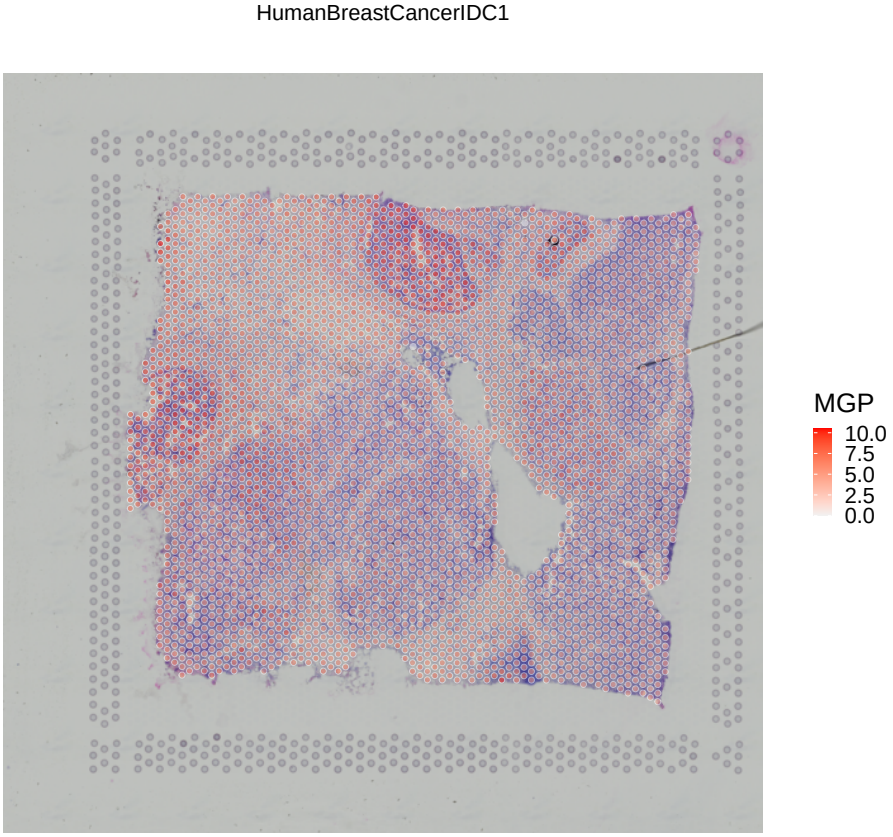


Fig. 9 Spatial expression of a highly variable gene

```
library(SpatialFeatureExperiment)
library(SFEData)
jbr = JanesickBreastData("rep1")
jbr
## class: SpatialFeatureExperiment
## dim: 541 167782
## metadata(1): Samples
## assays(1): counts
## rownames(541): ABCC11 ACTA2 ... BLANK_0497 BLANK_0499
## rowData names(6): ID Symbol ... vars cv2
## colnames: NULL
## colData names(10): Sample Barcode ... nCounts nGenes
## reducedDimNames(0):
## mainExpName: NULL
```

```
## altExpNames(0):
## spatialCoords names(2) : x_centroid y_centroid
## imgData names(1): sample_id
##
## unit:
## Geometries:
## colGeometries: centroids (POINT), cellSeg (POLYGON),
##      nucSeg (GEOMETRY)
##
## Graphs:
## sample01:
```

We can leverage the nature of in-situ data to explore the cell density across the tissue, identifying the tissue's macrostructure, and the cell segmentation, zooming in on a small portion of the tissue.

```
library(Voyager)
cellbins <- plotCellBin2D(jbr, hex = TRUE)
cellgeo <- plotGeometry(jbr, "cellSeg",
  bbox=c("xmin"=0, "ymin"=4000, "xmax"=1000, "ymax"=5000))
library(gridExtra)
grid.arrange(cellbins, cellgeo, ncol=2)
```

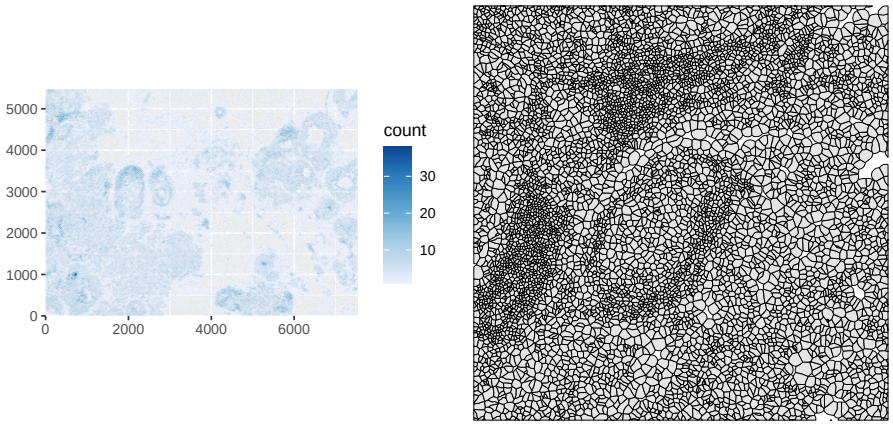


Fig. 10 Cell density and cell boundaries of a Xenium breast cancer sample

Finally, we can visualize the expression of marker genes after log-normalizing the data.

```
jbr <- jbr[, jbr$nCounts >= 20]
jbr <- logNormCounts(jbr)
library(scattermore)
strom <- plotSpatialFeature(jbr, "POSTN",
                 colGeometryName = "centroids",
scattermore = TRUE, ncol = 2, pointsize = 0.5) +
                 ggtitle("POSTN, stromal")
fasn <- plotSpatialFeature(jbr, "FASN",
                 colGeometryName = "centroids",
scattermore = TRUE, ncol = 2, pointsize = 0.5) +
ggtitle("FASN, invasive")
grid.arrange(strom, fasn, ncol=2)
```

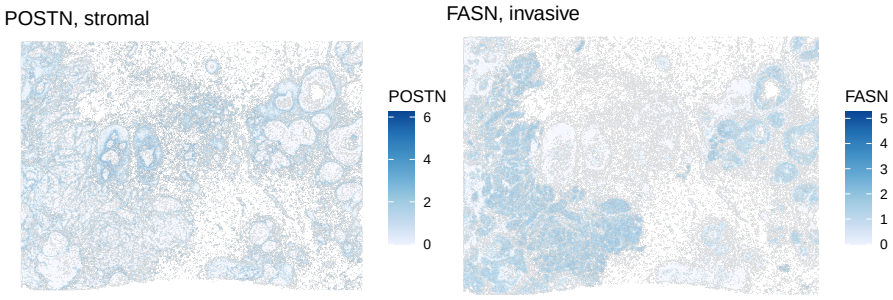


Fig. 11 Spatial expression of marker genes

6 Components and processes for introducing new data, analytic tools, documents

6.1 Contributions and review

Proposed contributions to Bioconductor's ecosystem of software packages, data resources, and documentation are registered at

`https://github.com/bioconductor/contributions/issues`

Contributors identify a public github.com repository that houses their software, or some durable open data repository for a data contribution. The contributor provides schematized information on format, licensing, and commitment to maintenance of the contributed resource. After a series of automated and manual verification steps, the contributed resource enters the review process.

An example under review in December 2023 is the "methodical" package, submitted 27 September 2023. The issue number at the contributions site is 3169. This contribution is of particular interest as it addresses new data resources from whole genome and reduced representation bisulfite sequencing experiments. Specifics on these high-resolution studies of DNA methylation in a variety of clinical situtions are given below.

6.2 Data structures

Inheritance is a key feature of object-oriented programming (OOP) that allows us to define a new class out of existing classes and add new features, which provides reusability of code. Inheritance carries over attributes and methods defined for base classes; 'Attributes' are variables that are bound in a class. They are used to define behavior and methods for objects of that class. 'Methods' are functions defined within a class that receive an instance of the class, conventionally called self, as the first argument. The attributes defined for a base class will automatically be present in the derived class, and the methods for the base class will work for the derived class. The R programming language has three different class systems: S3, S4, and Reference. Inheritance in S3 classes does not have any fixed definition, and hence attributes of S3 objects can be arbitrary. Derived classes, however, inherit the methods defined for the base class. Inheritance in S4 classes is more structured, and derived classes inherit both attributes and methods of the parent class. Reference classes are similar to S4 classes, but they are mutable and have reference semantics.

S4 classes are used extensively in Bioconductor to create data structures that store complex information, such as biological assay data and metadata, in one or more slots. The entire structure can then be assigned to an R object, and the types of information in each slot of the object are tightly controlled. S4 generics and methods define functions that can be applied to these objects, providing a rich software development infrastructure while ensuring interoperability, reusability, and efficiency.

Bioconductor have established Bioconductor classes to represent different types of biological data. Data and tools distributed through Bioconductor adopt Bioconductor classes, providing convenient methods and improving usability and interoperability within the Bioconductor ecosystem.

The GRanges class represents a collection of genomic ranges and associated annotations. Each element in the vector represents a set genomic ranges in terms of the sequence name (seqnames, typically the chromosome), start and end coordinates (ranges, as an IRanges object), strand (strand, either positive, negative, or

Table 7 Overview of key datatypes and associated classes in Bioconductor.

Data Types	Bioconductor Classes
Genomic coordinates (1-based, closed interval)	GRanges
Groups of genomic coordinates	GRangesList
Ragged genomic coordinates	RaggedExperiment
Gene sets	GeneSet
Rectangular Features x samples	SummarizedExperiment
Multi-omics data	MultiAssayExperiment
Single-cell data	SingleCellExperiment
Spatial Transcriptomics	SpatialExperiment
Mass spectrometry data	Spectra

unstranded), and optional metadata columns (e.g., exon_id and exon_name in the below).

```
GRanges object with 4 ranges and 2 metadata columns:
      seqnames            ranges strand |   exon_id       exon_name
         <Rle>         <IRanges>  <Rle> | <integer>     <character>
  [1]        X 99883667-99884983      - |    667145 ENSE00001459322
  [2]        X 99885756-99885863      - |    667146 ENSE00000868868
  [3]        X 99887482-99887565      - |    667147 ENSE00000401072
  [4]        X 99887538-99887565      - |    667148 ENSE00001849132
  -------
  seqinfo: 722 sequences (1 circular) from an unspecified genome
```

The GRangesList object serves as a container for genomic features consisting of multiple ranges that are grouped by a parent features, such as spliced transcripts that are comprised of exons. A GRangesList object behaves like a list and many of the same methods for GRanges objects are available for GRangesList object as well.

The SummarizedExperiment class (see Figure 1 is a matrix-like container, where rows represent features of interest (e.g., genes, transcripts, exons, etc.) and columns represent samples. The attributes of this object include experimental results (in assays), information on observations (in rowData) and samples (in colData), and additional metadata (in metadata). SummarizedExperiment objects can simultaneouly manage several experimental results as long as they are of the same dimensions. The best benefit of using SummarizedExperiment class is the coordination of the metadata and assays when subsetting. SummarizedExperiment is similar to the historical ExpressionSet class, but more flexible in its row information, allowing both GRanges and DataFrames. ExpressionSet object can be easily converted to SummarizedExperiment.

RangedSummarizedExperiment inherits the SummarizedExperiment class, with the extended capability of storing genomic ranges (as a GRanges or GRangesList object) of interest instead of a DataFrame (S4-class objectcs similar to data.frame) of features in rows.

The MultiAssayExperiment class (presented above in Figure 2) is modeled after the SummarizedExperiment class. A MultiAssayExperiment instance `M` can be filtered as a three-dimensional array. When `G` is a vector of feature identifiers, `C` a vector of sample identifiers, and `E` a vector of experiment names, then `M[G, C, E]` is a MultiAssayExperiment with content restricted to the requested features, samples, and experiments. The MultiAssayExperiment package includes tooling to convert data content to "long" or "wide" formats. In long format, each element of the assay array occupies a row, accompanied by metadata associated with the element. In wide format, each sample occupies a row, accompanied by all assocated assay and metadata elements.

6.3 Out-of-memory data representation strategies

We return to the "methodical" package submission mentioned above. A number of whole-genome bisulfite sequencing experiments on tumors from various anatomic sites are available in ExperimentHub. Metadata in that package shows that the datasets are large, ranging from 2-40 gigabytes. One smaller dataset is provided for illustration.

```
library(TumourMethData)
demm = download_meth_dataset("mcrpc_wg ..." ... [TRUNCATED]
demm
## class: RangedSummarizedExperiment
## dim: 1333114 100
## metadata(5): genome is_h5 ref_CpG chrom_sizes
##      descriptive_stats
## assays(2): beta cov
## rownames: NULL
## rowData names(0):
## colnames(100): DTB_003 DTB_005 ... DTB_265 DTB_266
## colData names(4): metastatis_site subtype age sex
rowRanges(demm)
## GRanges object with 1333114 ranges and 0 metadata columns:
##             seqnames    ranges strand
##                <Rle> <IRanges>  <Rle>
##         [1]    chr11     60077      *
##         [2]    chr11     60088      *
##         [3]    chr11     60365      *
##         [4]    chr11     60941      *
##         [5]    chr11     60979      *
##         ...      ...       ...    ...
##   [1333110]    chr11 135076482      *
```

```
##   [1333111]    chr11 135076496      *
##   [1333112]    chr11 135076502      *
##   [1333113]    chr11 135076507      *
##   [1333114]    chr11 135076510      *
##   -------
##   seqinfo: 25 sequences from an unspecified genome; no seqlengths
names(colData(demm))
## [1] "metastatis_site" "subtype"         "age"             "sex"
table(demm$metastatis_site)
##      Bone      Liver Lymph_node      Other
##        43         11         38          8
```

References to demm involve an 800MB excerpt of a prostate cancer atlas with a storage footprint of 40GB. Ideally, queries about particular genomic regions on particular samples, whole-sample statistical summaries, and searches for patterns can be carried out without specific accommodation of the data size or representation. The DelayedArray package helps pursue this aim. We'll illustrate by interrogating the prostate cancer WGBS data for "beta" (fraction of locus that is methylated) values in the vicinity of gene ATM.

```
library(EnsDb.Hsapiens.v86)
gg = genes(EnsDb.Hsapiens.v86)
# get gene addresses
atmpos = gg[gg$gene_name == "ATM" &
gg$gene_biotype == "protein_coding"] # filter to ATM
seqlevelsStyle(atmpos) = "UCSC"
assay(subsetByOverlaps(demm, atmpos+1e6))
## <18110 x 100> DelayedMatrix object of type "double":
##          DTB_003 DTB_005 DTB_008 ... DTB_265 DTB_266
##     [1,]  0.1053  0.7660  0.9206   .  0.6944  0.9412
##     [2,]  0.4062  0.9091  0.9318   .  0.5676  1.0000
##     [3,]  0.1379  0.0000  0.7400   .  0.4643  0.9231
##     [4,]  0.2308  0.9231  0.9149   .  0.8929  0.9286
##     [5,]  0.1481  0.8500  0.8864   .  0.8710  0.9762
##      ...       .       .       .   .       .       .
## [18106,]  0.4138  0.3143  0.3208   . 0.17647 0.10000
## [18107,]  0.2727  0.2745  0.4143   . 0.22500 0.32500
## [18108,]  0.2258  0.4800  0.5775   . 0.08889 0.25000
## [18109,]  0.5278  0.7059  0.8088   . 0.55263 0.97561
## [18110,]  0.2778  0.3137  0.6957   . 0.52632 0.35714
```

The numeric values presented above are just the "corners" of the associated array, presented as a "check" on the content requested. Transfer of array content to the CPU for numerical analysis only occurs on demand, which can be tailored to the quantity of RAM available at analysis time.

6.4 Quality assessment of Bioconductor resources

Figure 12 is an overview of the periodic ecosystem testing process for Bioconductor software packages in the release branch. All Bioconductor and CRAN packages on which they depend are present and are updated on change to sources.

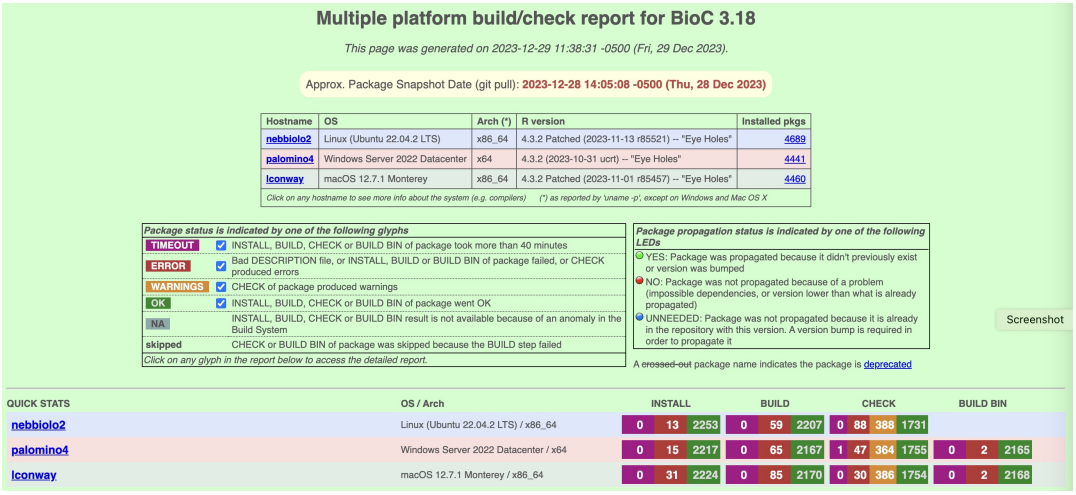


Fig. 12 Build report for Bioc 3.18, 12-29-2023.

The project distributes source tarballs for Linux-like systems, and compiled binaries for MacOS and Windows. Numbers in red boxes indicate failures to install, build, or check. Failure events are frequently platform-specific; full logs are provided on the build report pages to help developers isolate and fix build and check errors. When failures are persistent, developers are contacted by core. If contact cannot be made and failures continue, packages are deprecated for at least one release, and then removed.

7 Pedagogics and workforce development

The Bioconductor project has undertaken a number of initiatives to support growth of the scientific workforce's capacity to efficiently integrate and interpret genome-scale experiments.

- **Partnering with The Carpentries.** The Carpentries (https://carpentries.org) is a non-profit organization focused on teaching programming and data science to researchers. The organization defines "good practices in lesson design and development, and open source collaboration skills". Bioconductor community members have created bioc-intro, bioc-project, and bioc-rnaseq repositories using The Carpentries Incubator template. This arrangement helps Bioconductor create and manage a "train the trainer" process according to tested pedagogical principles.
- **Curating monographs for topics in genomic data science.** The breadth of Bioconductor resources for genomics, combined with the energetic approach to software and annotation upkeep in the project, empowers Bioconductor developers to produce unified, wide-ranging, computable documents on topics of interest to the broader cancer genomics community. Books currently available at bioconductor.org include OSCA (Orchestrating Single Cell Analysis with Bioconductor), SingleRBook (Assigning cell types with SingleR), csawBook (Analysis of ChIP-seq data), OHCA (Orchestrating Hi-C Analysis with Bioconductor) and R for Mass Spectrometry. Very recently, Jacques Serizay of Institut Pasteur has contributed a book authoring framework called BiocBook. This transforms documents marked up in Posit's quarto format into web-based books backed up by Docker containers and maintained with templated GitHub actions. The OHCA book is produced and managed with BiocBook.
- **A system for authoring and deploying interactive workshops.**

Figure 13 gives an overview of the resources and objectives of the system underlying workshop.bioconductor.org. Given a kubernetes-enabled cluster the workshop system assembles

- compute and storage elements,
- static components (training texts and shareable data),
- development environments (containers with all runtime elements required to compiled code, conduct analyses, communicate with GPUs).

A lightly customized deployment of the Galaxy system (usegalaxy.org) is used to deal with authentication and process initiation and termination.

This system has been used to serve interactive workshops in a number of international conferences. Content in R markdown or quarto can be produced by anyone interested in offering a workshop, and the "BiocWorkshopSubmit" app at workshop.bioconductor.org can be used to identify new content to the system. Markdown documents will be analyzed to determine what resources are needed for the containerization of workshop software and data components, and the container will be

Fig. 13 Workshop.bioconductor.org schematic.

created and registered at the GitHub Container Registry. Arrangements to deploy the workshop over a given calendar period can be made with Bioconductor core. The workshop container can be used to conduct the workshop on any system with a Docker client.

8 Conclusions and paths forward

We have described several aspects of Bioconductor's approach to ecosystem management for cancer genomics data science resources. In light of the dynamism of biotechnological innovation, it is clear that the project must anticipate change. But it is challenging to introduce changes to processes on which a very large community depends for their daily research work. Commitments to supporting reproducible research entail that Bioconductor preserves decades worth of images of software and data for immediate retrieval via web request by parties unknown to the project.

We'll conclude this report with a few observations on general paths that the project is likely to take that should have favorable consequences to researchers in cancer genomics.

- **Language-agnostic data and annotation** The `alabaster.*` packages introduced in Bioconductor 3.17 are designed to convert existing Bioconductor data structures to formats that are more readily ingested by software in other languages.

Thus the `alabaster.mae` package will convert a MultiAssayExperiment into a collection of files of metadata (serialized in JSON), sample-level data (serialized as CSV), and assay data (serialized to HDF5).

- **Zero-configuration genomic analysis environments** Users of Docker containers have long been able to take advantage of Bioconductor containers pre-populated with Rstudio and runtime resources to support installation of any desired software packages. The bioc2u system (https://github.com/bioconductor/bioc2u) in conjunction with r2u (github.com/eddelbuettel/r2u) introduces the availability of Debian packages for all Bioconductor packages, made available in a CRAN-like repository. Given a system running Ubuntu 22 or 20, the apt package manager will resolve any package requests with tested, fully linked binary packages. Users do not have to perform any configuration or compilation of system utilities or package code. This practice can greatly reduce resource consumption that occurs when individuals or workflow systems need to compile every package and its dependencies to perform analyses.
- **Computation at the data** Several members of Bioconductor's development core are on the technical development team of NHGRI's Analysis and Visualization Laboratory (AnVIL). The aim of this project is to overthrow the prevalent model of downloading data for local analysis. AnVIL mobilizes commercial cloud computing and storage to support truly elastic genomic analysis – create and pay for only the computation you need. The basic strategy is described in Schatz et al. [28].used in the production of the Telomere-to-Telomere genome build, see Aganezov et al. [29].

We hope that the project can continue to support researchers in cancer genomics for another 20 years!

9 Figure 7 software

Package	Version	Date(UTC)	Source
abind	1.4-5	2016-07-21	RSPM (R 4.2.0)
AnnotationDbi	1.64.1	2023-11-03	Bioconductor
AnnotationHub	3.10.0	2023-10-24	Bioconductor
backports	1.4.1	2021-12-13	RSPM (R 4.2.0)
bcellViper	1.38.0	2023-10-26	Bioconductor
Biobase	2.62.0	2023-10-24	Bioconductor
BiocFileCache	2.10.1	2023-10-26	Bioconductor
BiocGenerics	0.48.1	2023-11-01	Bioconductor
BiocManager	1.30.22	2023-08-08	RSPM (R 4.2.0)
BiocParallel	1.36.0	2023-10-24	Bioconductor
BiocVersion	3.18.0	2023-04-25	Bioconductor
Biostrings	2.70.1	2023-10-25	Bioconductor
bit	4.0.5	2022-11-15	RSPM (R 4.2.0)
bit64	4.0.5	2020-08-30	RSPM (R 4.2.0)
bitops	1.0-7	2021-04-24	RSPM (R 4.2.0)
blob	1.2.4	2023-03-17	RSPM (R 4.2.0)
broom	1.0.5	2023-06-09	RSPM (R 4.2.0)
bspm	0.5.5	2023-08-22	CRAN (R 4.3.1)
cachem	1.0.8	2023-05-01	RSPM (R 4.2.0)
car	3.1-2	2023-03-30	RSPM (R 4.2.0)
carData	3.0-5	2022-01-06	RSPM (R 4.2.0)
class	7.3-22	2023-05-03	RSPM (R 4.2.0)
cli	3.6.2	2023-12-11	RSPM (R 4.3.0)
codetools	0.2-19	2023-02-01	RSPM (R 4.2.0)
coin	1.4-3	2023-09-27	RSPM (R 4.3.0)
colorspace	2.1-0	2023-01-23	RSPM (R 4.2.0)
cowplot	1.1.2	2023-12-15	RSPM (R 4.3.0)
crayon	1.5.2	2022-09-29	RSPM (R 4.2.0)
curl	5.2.0	2023-12-08	RSPM (R 4.3.0)
DBI	1.1.3	2022-06-18	RSPM (R 4.2.0)
dbplyr	2.4.0	2023-10-26	RSPM (R 4.3.0)
decoupleR	2.8.0	2023-10-24	Bioconductor
DelayedArray	0.28.0	2023-10-24	Bioconductor
DESeq2	1.42.0	2023-10-24	Bioconductor
digest	0.6.33	2023-07-07	RSPM (R 4.2.0)
dorothea	1.14.0	2023-10-26	Bioconductor
dplyr	1.1.4	2023-11-17	RSPM (R 4.3.0)
e1071	1.7-14	2023-12-06	RSPM (R 4.3.0)
easier	1.8.0	2023-10-24	Bioconductor
easierData	1.8.0	2023-10-26	Bioconductor
ellipsis	0.3.2	2021-04-29	RSPM (R 4.2.0)
evaluate	0.23	2023-11-01	RSPM (R 4.3.0)
ExperimentHub	2.10.0	2023-10-24	Bioconductor
fansi	1.0.6	2023-12-08	RSPM (R 4.3.0)

farver	2.1.1	2022-07-06	RSPM (R 4.2.0)
fastmap	1.1.1	2023-02-24	RSPM (R 4.2.0)
filelock	1.0.3	2023-12-11	RSPM (R 4.3.0)
generics	0.1.3	2022-07-05	RSPM (R 4.2.0)
GenomeInfoDb	1.38.1	2023-11-08	Bioconductor
GenomeInfoDbData	1.2.11	¡NA¿	Bioconductor
GenomicRanges	1.54.1	2023-10-29	Bioconductor
ggplot2	3.4.4	2023-10-12	RSPM (R 4.3.0)
ggpubr	0.6.0	2023-02-10	RSPM (R 4.2.0)
ggrepel	0.9.4	2023-10-13	RSPM (R 4.3.0)
ggsignif	0.6.4	2022-10-13	RSPM (R 4.2.0)
glue	1.6.2	2022-02-24	RSPM (R 4.2.0)
gridExtra	2.3	2017-09-09	RSPM (R 4.2.0)
gtable	0.3.4	2023-08-21	RSPM (R 4.2.0)
htmltools	0.5.7	2023-11-03	RSPM (R 4.3.0)
htmlwidgets	1.6.4	2023-12-06	RSPM (R 4.3.0)
httpuv	1.6.13	2023-12-06	RSPM (R 4.3.0)
httr	1.4.7	2023-08-15	RSPM (R 4.2.0)
interactiveDisplayBase	1.40.0	2023-10-24	Bioconductor
IRanges	2.36.0	2023-10-24	Bioconductor
jsonlite	1.8.8	2023-12-04	RSPM (R 4.3.0)
KEGGREST	1.42.0	2023-10-24	Bioconductor
kernlab	0.9-32	2023-01-31	RSPM (R 4.2.0)
KernSmooth	2.23-22	2023-07-10	RSPM (R 4.2.0)
knitr	1.45	2023-10-30	RSPM (R 4.3.0)
labeling	0.4.3	2023-08-29	RSPM (R 4.2.0)
later	1.3.2	2023-12-06	RSPM (R 4.3.0)
lattice	0.22-5	2023-10-24	RSPM (R 4.3.0)
lazyeval	0.2.2	2019-03-15	RSPM (R 4.2.0)
libcoin	1.0-10	2023-09-27	RSPM (R 4.3.0)
lifecycle	1.0.4	2023-11-07	RSPM (R 4.3.0)
limSolve	1.5.7	2023-09-21	RSPM (R 4.3.0)
locfit	1.5-9.8	2023-06-11	RSPM (R 4.2.0)
lpSolve	5.6.20	2023-12-10	RSPM (R 4.3.0)
magrittr	2.0.3	2022-03-30	RSPM (R 4.2.0)
MASS	7.3-60	2023-05-04	RSPM (R 4.2.0)
Matrix	1.6-4	2023-11-30	RSPM (R 4.3.0)
MatrixGenerics	1.14.0	2023-10-24	Bioconductor
matrixStats	1.2.0	2023-12-11	RSPM (R 4.3.0)
memoise	2.0.1	2021-11-26	RSPM (R 4.2.0)
mime	0.12	2021-09-28	RSPM (R 4.2.0)
mixtools	2.0.0	2022-12-05	RSPM (R 4.2.0)
modeltools	0.2-23	2020-03-05	RSPM (R 4.2.0)
multcomp	1.4-25	2023-06-20	RSPM (R 4.2.0)
munsell	0.5.0	2018-06-12	RSPM (R 4.2.0)

mvtnorm	1.2-4	2023-11-27 RSPM (R 4.3.0)
nlme	3.1-164	2023-11-27 RSPM (R 4.3.0)
pillar	1.9.0	2023-03-22 RSPM (R 4.2.0)
pkgconfig	2.0.3	2019-09-22 RSPM (R 4.2.0)
plotly	4.10.3	2023-10-21 RSPM (R 4.3.0)
plyr	1.8.9	2023-10-02 RSPM (R 4.3.0)
png	0.1-8	2022-11-29 RSPM (R 4.2.0)
preprocessCore	1.64.0	2023-10-24 Bioconductor
progeny	1.24.0	2023-10-24 Bioconductor
promises	1.2.1	2023-08-10 RSPM (R 4.2.0)
proxy	0.4-27	2022-06-09 RSPM (R 4.2.0)
purrr	1.0.2	2023-08-10 RSPM (R 4.2.0)
quadprog	1.5-8	2019-11-20 RSPM (R 4.2.0)
quantiseqr	1.10.0	2023-10-24 Bioconductor
R6	2.5.1	2021-08-19 RSPM (R 4.2.0)
rappdirs	0.3.3	2021-01-31 RSPM (R 4.2.0)
Rcpp	1.0.11	2023-07-06 RSPM (R 4.2.0)
RCurl	1.98-1.13	2023-11-02 RSPM (R 4.3.0)
reshape2	1.4.4	2020-04-09 CRAN (R 4.0.1)
rlang	1.1.2	2023-11-04 RSPM (R 4.3.0)
rmarkdown	2.25	2023-09-18 RSPM (R 4.3.0)
ROCR	1.0-11	2020-05-02 RSPM (R 4.2.0)
RSQLite	2.3.4	2023-12-08 RSPM (R 4.3.0)
rstatix	0.7.2	2023-02-01 RSPM (R 4.2.0)
S4Arrays	1.2.0	2023-10-24 Bioconductor
S4Vectors	0.40.2	2023-11-23 Bioconductor 3.18 (R 4.3.2)
sandwich	3.1-0	2023-12-11 RSPM (R 4.3.0)
scales	1.3.0	2023-11-28 RSPM (R 4.3.0)
segmented	2.0-1	2023-12-19 RSPM (R 4.3.0)
sessioninfo	1.2.2	2021-12-06 RSPM (R 4.2.0)
shiny	1.8.0	2023-11-17 RSPM (R 4.3.0)
SparseArray	1.2.2	2023-11-07 Bioconductor
startup	0.21.0	2023-12-11 RSPM (R 4.3.0)
stringi	1.8.3	2023-12-11 RSPM (R 4.3.0)
stringr	1.5.1	2023-11-14 RSPM (R 4.3.0)
SummarizedExperiment	1.32.0	2023-10-24 Bioconductor
survival	3.5-7	2023-08-14 RSPM (R 4.2.0)
TH.data	1.1-2	2023-04-17 RSPM (R 4.2.0)
tibble	3.2.1	2023-03-20 RSPM (R 4.3.0)
tidyr	1.3.0	2023-01-24 RSPM (R 4.2.0)
tidyselect	1.2.0	2022-10-10 RSPM (R 4.2.0)
utf8	1.2.4	2023-10-22 RSPM (R 4.3.0)
vctrs	0.6.5	2023-12-01 RSPM (R 4.3.0)
viper	1.36.0	2023-10-24 Bioconductor
viridisLite	0.4.2	2023-05-02 RSPM (R 4.2.0)
withr	2.5.2	2023-10-30 RSPM (R 4.3.0)
xfun	0.41	2023-11-01 RSPM (R 4.3.0)
xtable	1.8-4	2019-04-21 RSPM (R 4.2.0)
XVector	0.42.0	2023-10-24 Bioconductor
yaml	2.3.8	2023-12-11 RSPM (R 4.3.0)
zlibbioc	1.48.0	2023-10-24 Bioconductor
zoo	1.8-12	2023-04-13 RSPM (R 4.2.0)

10 Acknowledgments

This work was supported in part by NIH NCI 3U24CA180996-10S1, NHGRI 5U24HG004059-18, and NSF ACCESS allocation BIR190004.

References

1. R-Core. *Writing R Extensions*, 2024.
2. Aevermann, B. D., Novotny, M., Bakken, T., Miller, J. A., Diehl, A. D., Osumi-Sutherland, D., Lasken, R. S., Lein, E. S., and Scheuermann, R. H. Cell type discovery using single-cell transcriptomics: Implications for ontological representation. *Human Molecular Genetics*, 27:R40–R47, 2018.
3. Tian, L., Su, S., Dong, X., Amann-Zalcenstein, D., Biben, C., Seidi, A., Hilton, D. J., Naik, S. H., and Ritchie, M. E. scpipe: A flexible r/bioconductor preprocessing pipeline for single-cell rna-sequencing data. *PLOS Computational Biology*, 14(8):e1006361, 2018.
4. Wei, Z., Zhang, W., Fang, H., Li, Y., and Wang, X. esatac: An easy-to-use systematic pipeline for atac-seq data analysis. *Bioinformatics (Oxford, England)*, March 2018.
5. Ou, J., Liu, H., Yu, J., Kelliher, M. A., Castilla, L. H., Lawson, N. D., and Zhu, L. J. Atacseqqc: a bioconductor package for post-alignment quality assessment of atac-seq data. *BMC Genomics*, 19(1), 2018.
6. Aryee, M. J., Jaffe, A. E., Corrada-Bravo, H., Ladd-Acosta, C., Feinberg, A. P., Hansen, K. D., and Irizarry, R. A. Minfi: a flexible and comprehensive bioconductor package for the analysis of infinium dna methylation microarrays. *Bioinformatics*, 30(10):1363–1369, 2014.
7. Hansen, K. D., Langmead, B., and Irizarry, R. A. Bsmooth: from whole genome bisulfite sequencing reads to differentially methylated regions. *Genome Biology*, 13(10):R83, 2012.
8. Müller, F., Scherer, M., Assenov, Y., Lutsik, P., Walter, J., Lengauer, T., and Bock, C. Rnbeads 2.0: comprehensive analysis of dna methylation data. *Genome Biology*, 20(1), 2019.
9. Lawson, J., Tomazou, E., Bock, C., and Sheffield, N. C. Mira: An R package for DNA methylation-based inference of regulatory activity. *Bioinformatics*, bty083, 3 2018.
10. Silva, T. C., Coetzee, S. G., Gull, N., Yao, L., Hazelett, D. J., Noushmehr, H., Lin, D.-C., and Berman, B. P. Elmer v.2: an r/bioconductor package to reconstruct gene regulatory networks from dna methylation and transcriptome profiles. *Bioinformatics*, 35(11):1974–1977, 2019.
11. Zheng, S. C., Breeze, C. E., Beck, S., and Teschendorff, A. E. Identification of differentially methylated cell types in epigenome-wide association studies. *Nature Methods*, 15(12):1059–1066, 2018.
12. Stark, R. and Brown, G. *DiffBind: differential binding analysis of ChIP-Seq peak data*, 2011.
13. Kupkova, K., Mosquera, J. V., Smith, J. P., Stolarczyk, M., Danehy, T. L., Lawson, J. T., Xue, B., Stubbs, J. T., LeRoy, N., and Sheffield, N. C. GenomicDistributions: fast analysis of genomic intervals with bioconductor. *BMC Genomics*, 23(1), apr 2022.
14. Argelaguet, R., Arnol, D., Bredikhin, D., Deloro, Y., Velten, B., Marioni, J. C., and Stegle, O. Mofa+: a statistical framework for comprehensive integration of multi-modal single-cell data. *Genome Biology*, 21(1), 2020.
15. Lawson, J. T., Smith, J. P., Bekiranov, S., Garrett-Bakelman, F. E., and Sheffield, N. C. COCOA: coordinate covariation analysis of epigenetic heterogeneity. *Genome Biology*, 21(1), sep 2020.
16. Sheffield, N. C. and Bock, C. Lola: enrichment analysis for genomic region sets and regulatory elements in R and bioconductor. *Bioinformatics*, 32(4):587–589, Oct 2016.
17. Gel, B., Diez-Villanueva, A., Serra, E., Buschbeck, M., Peinado, M. A., and Malinverni, R. regioneR: an r/bioconductor package for the association analysis of genomic regions based on permutation tests. *Bioinformatics*, page btv562, sep 2015.

18. Aibar, S., Fontanillo, C., Droste, C., and De Las Rivas, J. Functional gene networks: R/bioc package to generate and analyse gene networks derived from functional enrichment and clustering. *Bioinformatics*, 31(10):1686–1688, 2015.
19. Zhu, L. J., Gazin, C., Lawson, N. D., Pagès, H., Lin, S. M., Lapointe, D. S., and Green, M. R. ChIPpeakAnno: a bioconductor package to annotate ChIP-seq and ChIP-chip data. *BMC Bioinformatics*, 11(1), may 2010.
20. Cavalcante, R. G. and Sartor, M. A. annotatr: genomic regions in context. *Bioinformatics*, 33(15):2381–2383, mar 2017.
21. Immune checkpoint inhibitors. https://www.cancer.gov/about-cancer/treatment/types/immunotherapy/checkpoint-inhibitors, 2022. Accessed: 2023-12-30.
22. Óscar Lapuente-Santana, van Genderen, M., Hilbers, P. A., Finotello, F., and Eduati, F. Interpretable systems biomarkers predict response to immune-checkpoint inhibitors. *Patterns*, 2, 8 2021.
23. Mariathasan, S., Turley, S. J., Nickles, D., Castiglioni, A., Yuen, K., Wang, Y., Kadel III, E. E., Koeppen, H., Astarita, J. L., Cubas, R., Jhunjhunwala, S., Banchereau, R., Yang, Y., Guan, Y., Chalouni, C., Ziai, J., Senbabaoglu, Y., Santoro, S., Sheinson, D., Hung, J., Giltnane, J. M., Pierce, A. A., Mesh, K., Lianoglou, S., Riegler, J., Carano, R. A. D., Eriksson, P., Hoglund, M., Somarriba, L., Halligan, D. L., van der Heijden, M. S., Loriot, Y., Rosenberg, J. E., Fong, L., Mellman, I., Chen, D. S., Green, M., Derleth, C., Fine, G. D., Hegde, P. S., Bourgon, R., and Powles, T. Tgfb attenuates tumour response to pd-l1 blockade by contributing to exclusion of t cells. *Nature*, 554(7693):544–548, Feb 2018.
24. Righelli, D., Weber, L. M., Crowell, H. L., Pardo, B., Collado-Torres, L., Ghazanfar, S., Lun, A. T., Hicks, S. C., and Risso, D. Spatialexperiment: infrastructure for spatially-resolved transcriptomics data in r using bioconductor. *Bioinformatics*, 38(11):3128–3131, 2022.
25. Moses, L., Einarsson, P. H., Jackson, K., Luebbert, L., Booeshaghi, A. S., Antonsson, S., Bray, N., Melsted, P., and Pachter, L. Voyager: exploratory single-cell genomics data analysis with geospatial statistics. *bioRxiv*, 2023.
26. Couto, B. Z. P., Robertson, N., Patrick, E., and Ghazanfar, S. Moleculeexperiment enables consistent infrastructure for molecule-resolved spatial transcriptomics data in bioconductor. *bioRxiv*, 2023.
27. Weber, L. M., Saha, A., Datta, A., Hansen, K. D., and Hicks, S. C. nnsvg for the scalable identification of spatially variable genes using nearest-neighbor gaussian processes. *Nature communications*, 14(1):4059, 2023.
28. Schatz, M. C., Philippakis, A. A., Afgan, E., Banks, E., Carey, V. J., Carroll, R. J., Culotti, A., Ellrott, K., Goecks, J., Grossman, R. L., Hall, I. M., Hansen, K. D., Lawson, J., Leek, J. T., Luria, A. O., Mosher, S., Morgan, M., Nekrutenko, A., O'Connor, B. D., Osborn, K., Paten, B., Patterson, C., Tan, F. J., Taylor, C. O., Vessio, J., Waldron, L., Wang, T., Wuichet, K., Baumann, A., Rula, A., Kovalsy, A., Bernard, C., Caetano-Anollés, D., der Auwera, G. A. V., Canas, J., Yuksel, K., Herman, K., Taylor, M. M., Simeon, M., Baumann, M., Wang, Q., Title, R., Munshi, R., Chaluvadi, S., Reeves, V., Disman, W., Thomas, S., Hajian, A., Kiernan, E., Gupta, N., Vosburg, T., Geistlinger, L., Ramos, M., Oh, S., Rogers, D., McDade, F., Hastie, M., Turaga, N., Ostrovsky, A., Mahmoud, A., Baker, D., Clements, D., Cox, K. E., Suderman, K., Kucher, N., Golitsynskiy, S., Zarate, S., Wheelan, S. J., Kammers, K., Stevens, A., Hutter, C., Wellington, C., Ghanaim, E. M., Wiley, K. L., Sen, S. K., Francesco, V. D., s Yuen, D., Walsh, B., Sargent, L., Jalili, V., Chilton, J., Shepherd, L., Stubbs, B., O'Farrell, A., Vizzier, B. A., Overbeck, C., Reid, C., Steinberg, D. C., Sheets, E. A., Lucas, J., Blauvelt, L., Cabansay, L., Warren, N., Hannafious, B., Harris, T., Reddy, R., Torstenson, E., Banasiewicz, M. K., Abel, H. J., and Walker, J. Inverting the model of genomics data sharing with the nhgri genomic data science analysis, visualization, and informatics lab-space. *Cell Genomics*, 2:100085, 1 2022.
29. Aganezov, S., Yan, S. M., Soto, D. C., Kirsche, M., Zarate, S., Avdeyev, P., Taylor, D. J., Shafin, K., Shumate, A., Xiao, C., Wagner, J., McDaniel, J., Olson, N. D., Sauria, M. E., Vollger, M. R., Rhie, A., Meredith, M., Martin, S., Lee, J., Koren, S., Rosenfeld, J. A., Paten, B., Layer, R., Chin, C. S., Sedlazeck, F. J., Hansen, N. F., Miller, D. E., Phillippy, A. M., Miga, K. H., McCoy, R. C., Dennis, M. Y., Zook, J. M., and Schatz, M. C. A complete reference genome improves analysis of human genetic variation. *Science*, 376, 2022.

11 Figure captions

Figure 1 SummarizedExperiment schematic.

Figure 2 MultiAssayExperiment schematic.

Figure 3 Survival profile extraction from three MultiAssayExperiments produced with curatedTCGAData calls.

Figure 4 Survival distributions for donors of breast tumors in TCGA, stratified by presence or absence of mutation in gene TTN.

Figure 5 Density of recurrent genomic alterations on chromosome 17 for 48 angiosarcoma patients.

Figure 6 Ontology visualization and tabulation with ontoProc::ctmarks.

Figure 7 Comparison of genomic features distinguishing patients non-responsive and responsive to immune checkpoint blockade.

Figure 8 Visualization of a Visium breast cancer sample.

Figure 9 Spatial expression of a highly variable gene.

Figure 10 Cell density and cell boundaries of a Xenium breast cancer sample.

Figure 11 Spatial expression of marker genes.

Figure 12 Build report for Bioc 3.18, 12-29-2023.

Figure 13 Workshop.bioconductor.org schematic.